

Document Object Model (DOM), часть 1

JS
COURSE
ORT DNIPRO

ORTDNIPRO.ORG/JS

1. HTML Document

Структура HTML-документа

состоит из:

<tag attr="value">Text data</tag>

Теги как контейнер для информации
+ атрибуты

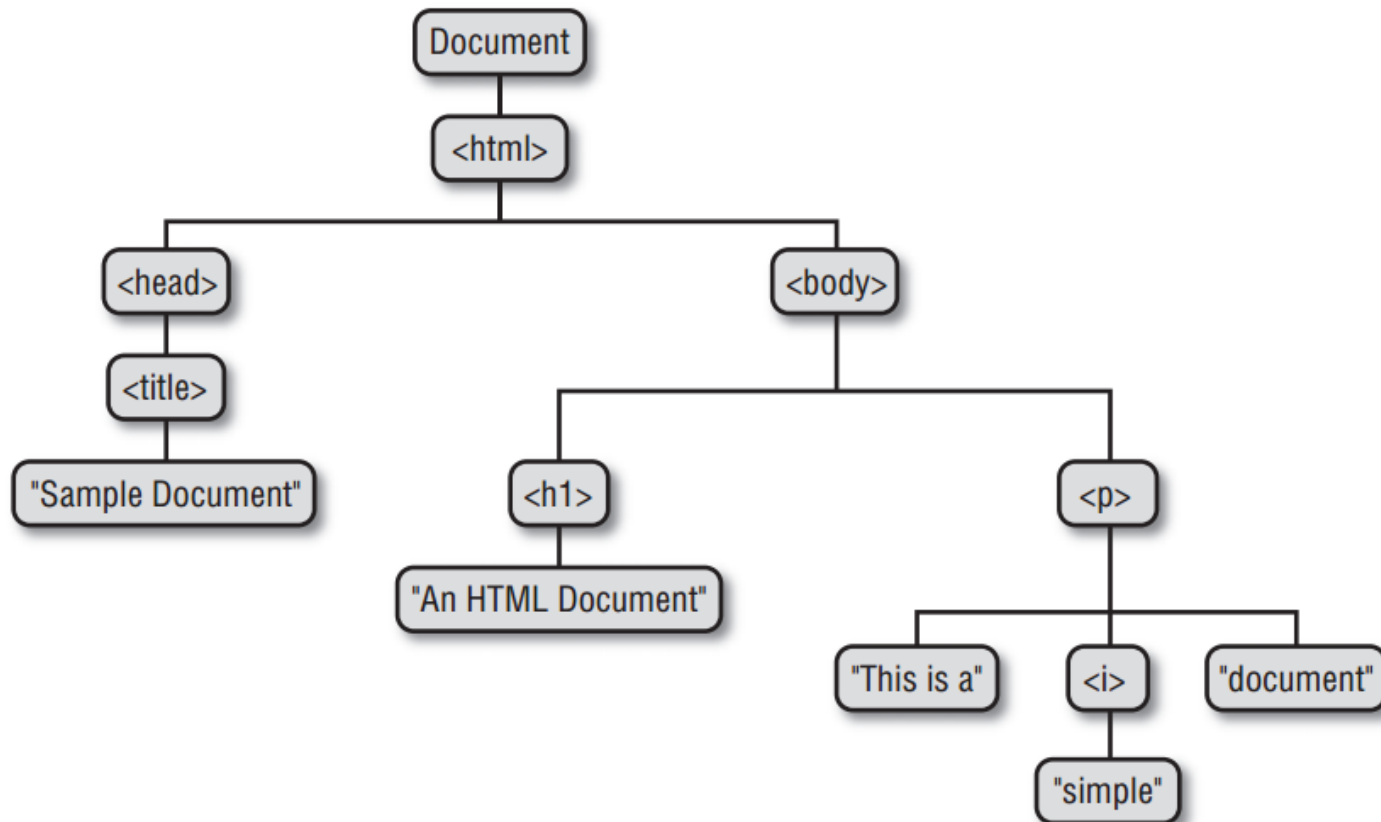
Текстовые данные (содержимое, контент)

Структура HTML-документа

```
1  <!DOCTYPE html>
2  <html lang="en">
3      <head>
4          <title>Sample Document</title>
5      </head>
6      <body>
7          <h1>An HTML Document</h1>
8          <p>This is <i>simple</i> document</p>
9      </body>
10 </html>
```

Древовидная структура HTML-документа

Древовидная структура HTML-документа



В контексте **JavaScript**, каждый **тег** дерева представлен **объектом** (часто используется термин: **узел, node**). У каждого элемента есть один **родительский элемент**, и множество **дочерних элементов** (от 0 до ∞).

2. DOM

Document Object Model (DOM)

Объектная Модель Документа

Стандарт определяющий из каких объектов браузер собирает дерево документа, и какие свойства и методы есть у этих объектов.

Задача JavaScript – манипуляция HTML-документом

1. Добавление нового элемента:

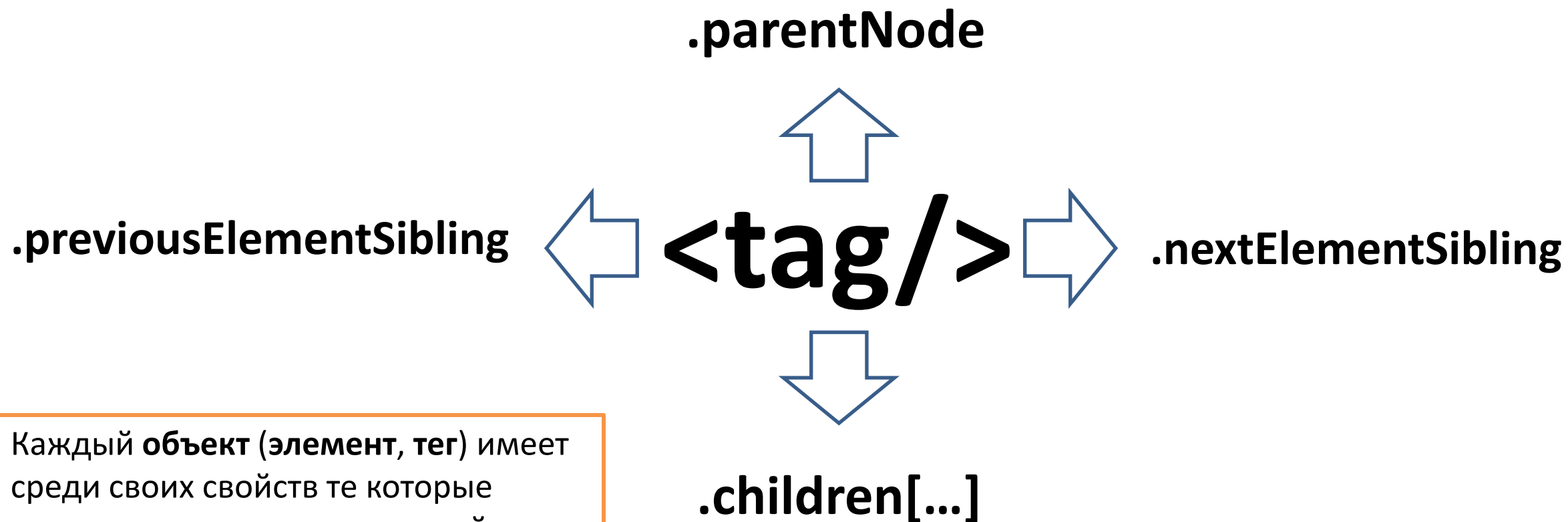
Создать новый элемент и присоединить его, в качестве дочернего, к одному из существующих элементов;

2. Изменение элемента:

*Изменение свойств элемента (в т.ч. содержимого);
Изменение его позиции в дереве документа;*

3. Удаление элемента (из дерева документа).

Свойства элементов HTML-документа



Каждый **объект (элемент, тег)** имеет среди своих свойств те которые хранят ссылку на родительский элемент (**parentNode**), на соседние элементы (**previousElementSibling** и **nextElementSibling**) и на перечень потомков (**childNodes** и **children**)

Свойства элементов HTML-документа

<tag/>

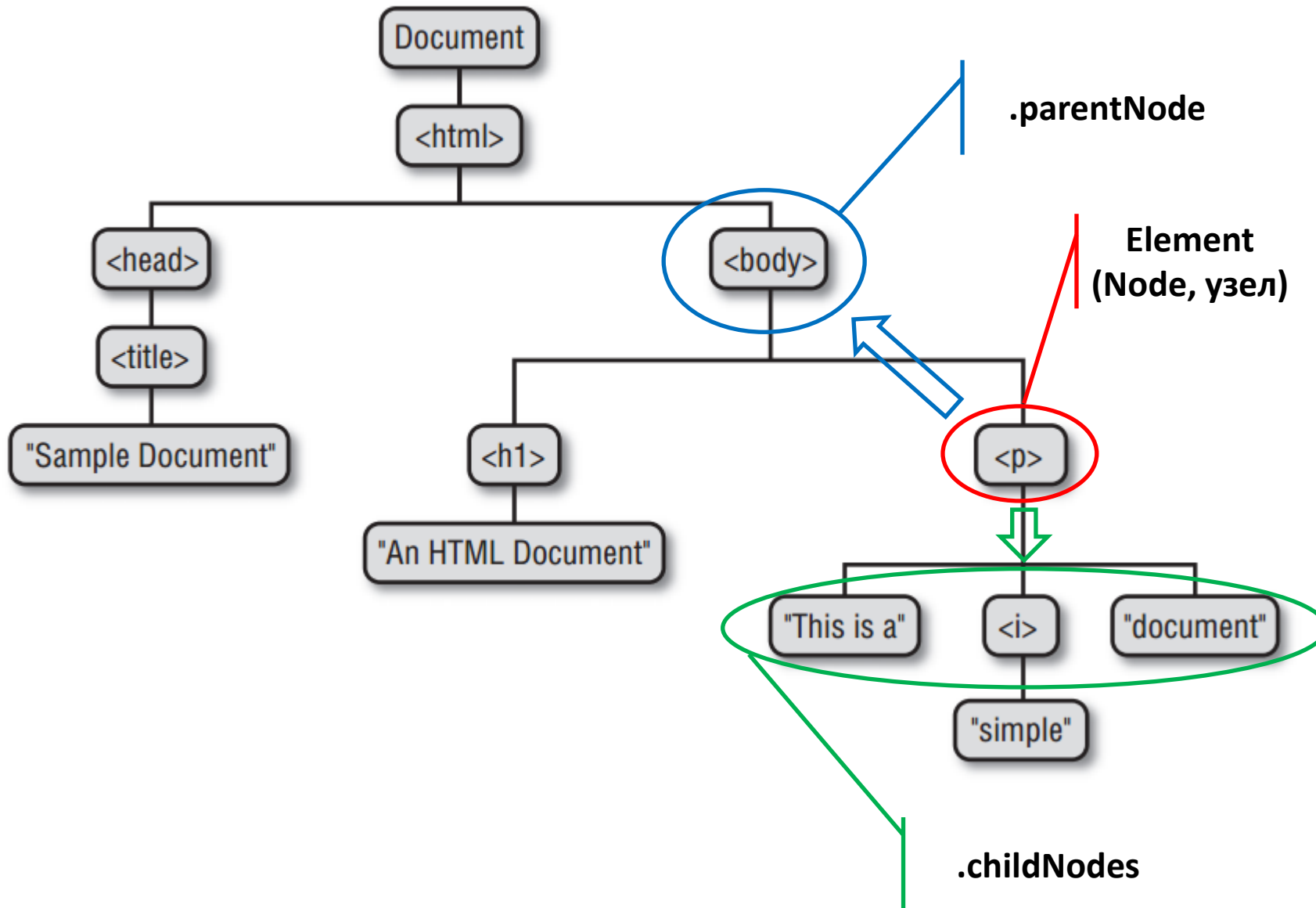
Также среди **свойств** объекта (**элемента, тега**) есть те которые позволяют управлять содержимым (**атрибутами, стилями**) или подпиской на **событиями**, а также ряд методов позволяющих добавлять/удалять элементы, и искать вложенные элементы.

.id
.innerHTML
.className
.classList[...]
.attributes[...]
.style { ... }

.onclick
.ondblclick
.onmouseenter

.appendChild()
.insertBefore()
.remove()
.insertAdjacentHTML()
.insertAdjacentElement()
.insertAdjacentText()
...

DOM – Document Object Model



globalThis.document
корень дерева документа (**window**)

globalThis.document.children – массив с тегами верхнего уровня.

**3. Как добраться
(найти) до тега?**

Теги у которых есть атрибут **id**
доступны сразу как переменные
ссылкой на объект

*Но только если **id** состоит из допустимых
для имён переменных в **JavaScript** символов.*

Поиск элементов в документе

Выбор элемента с которым проводить манипуляции самая часто выполняемая операция в JS.

Выбор элемента по атрибуту id:

```
document.getElementById("some_id") ;
```

*Возвращает один элемент атрибут (свойство) **id** равно «**some_id**». Если такого элемента нет в документе, то возвращается **null**.*

Поиск элементов в документе

Выбор элементов по названию тега:

```
document.getElementsByTagName ("tag_name") ;
```

Выбор элементов по атрибуту name:

```
document.getElementsByName ("attr_name") ;
```

Выбор элементов по атрибуту class:

```
document.getElementsByClassName ("class_name") ;
```

Все эти функции возвращают псевдомассив с теми элементами которые подошли под условие.

Поиск элементов в документе

Выбор всех элементов которые соответствуют CSS селектору:

```
document.querySelectorAll("css_selector") ;
```

Возвращает псевдомассив с теми элементами которые подошли под условие css-селектора.

```
document.querySelector("css_selector") ;
```

*Возвращает первый найденный элемент который подошел под условие css-селектора (или **null** если ничего не найдено).*

4. Как изменить тег?

Свойство **.innerHTML** хранит содержимое тега

Свойство **.innerHTML** – можно не только считывать но и устанавливать. Изменение свойства **.innerHTML** – автоматически влечёт перерисовку документа.

Полезные свойства элементов

.className – свойство содержит полный список всех классов которые присвоены тегу (одной строкой).

.classList – свойство содержит список всех классов которые присвоены тегу (в виде массива).

.classList.add('cat') – метод добавляет класс к тегу (если есть другие классы то они остаются).

.classList.remove('cat') – метод удаляет класс у тегу (если есть другие классы то они не затрагиваются).

.classList.toggle('cat') – метод удаляет класс у тегу, если он есть, или добавляет класс, если его нет.

.classList.contains('cat') – метод проверяет наличие у тега заданного класса (возвращает true/false).

.style – свойство определяющее объект со всеми поддерживаемыми браузером стилевые свойства (CSS).

.attributes – хранит коллекцию с атрибутами тега.

5. Как удалить тег?

Удаление элементов из дерева документа

```
54  
55     let tag = document.getElementById('special');  
56  
57     tag.remove();  
58  
59     console.dir(tag);  
60
```

Удалить элемент из дерева документа можно вызывая у него метод **.remove()**, при этом все его дочерние элементы также исчезнут со страницы. Однако сам объект-тег не уничтожается. Его можно использовать в дальнейшем.

```
► ownerDocument: document  
  parentElement: null  
  parentNode: null  
  prefix: null
```

6. Немного практики

Пересчитаем цены



Lorem, ipsum dolor sit amet consectetur adipisicing.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Ab nemo officia sit recusandae. Ex quos libero deleniti. Laborum vitae earum, fuga nihil, autem accusamus obcaecati eveniet repudiandae iure totam aspernatur perspiciatis harum illum perferendis magni, beatae voluptatibus quaerat? Perferendis alias laudantium odio provident saepe culpa voluptas corporis sed nihil voluptatibus consequatur, sapiente eum molestias, id reiciendis. Debitis officia quo, dolorem ea vitae earum labore maxime quisquam ipsa tenetur temporibus, neque perspiciatis! Cupiditate sint incidunt vero in repellat provident qui nisi ullam magnam porro, reiciendis beatae ea rem commodi corrupti deleniti reprehenderit, aliquam eos harum modi. Veniam maiores eius beatae modi.

Select Your Plan

Free	Pro	Enterprise
\$0/mo	\$15/mo	\$29/mo
10 users included 2 GB of storage Email support Help center access	20 users included 10 GB of storage Priority email support Help center access	30 users included 15 GB of storage Phone and email support Help center access
Sign up for free	Get started	Contact us

Lorem ipsum dolor sit amet consectetur adipisicing elit. Laborum modi adipisci tenetur, veniam mollitia quibusdam inventore minus doloribus non dolore animi, tempora architecto ab ipsum illo nobis. Itaque repudiandae eius error ab consectetur nisi? Velit quam exercitationem dolor quod ratione earum provident vel quia vero ipsa ipsam necessitatibus, similique reprehenderit est laborum voluptatibus sapiente architecto tenetur unde ut fugit natus rem? Nisi aperiam aspernatur maiores quasi voluptatum amet delectus quam obcaecati. Inventore sapiente possimus harum quisquam ratione fuga tenetur ullam voluptatem, in voluptatum beatae eveniet rerum autem architecto unde culpa laudantium laboriosam, nihil nulla voluptate consequatur! Debitis consequatur doloribus maiores.

Воспользуемся шаблоном [./src/pricing-template/](#) и пересчитаем цены в гривну, на основании курса НБУ.

<https://bank.gov.ua/NBUStatService/v1/statdirectory/exchange?json>

На следующем занятии

На следующем занятии

Работа с DOM

(работа с разметкой документа)

часть 2